Machine Code (cont.)

November 8

CSC201 Section 002

Fall, 2000

Instruction Encoding

- Instructions = 1 to 9 bytes long
- First byte = Opcode, or Opcode+Reg
- Second byte (if needed) = the ModRegR/M byte
  - Mod = 2 bits
  - Reg = 3 bits
  - R/M = 3 bits
- Third byte needed for based+indexed addressing mode = the SIB byte
  - S (scaling) = 2 bits
  - I (Index register) = 3 bits
  - B (Base register) = 3 bits
- Plus additional bytes for addresses (offsets) and immediate (constant) values

Encoding Instructions (Appendix C)

- Be familiar with the table on p. 320
  - Abbreviations for operand types
  - Notes on the Reg field of the ModRegR/M byte
  - Abbreviations for the additional fields (ib, iw, id, cd)
- Step 1: Look up the opcode / operand combination in the "Pentium Machine Language Table". Result:
  - First byte = opcode, in hex (+ add register offset, if any)
  - Note about the Reg field of the next (ModRegR/M) byte
  - Immediate value length (if any), or offset value (for jumps)

Encoding (cont.)

- Step 2: If there are no notes on the Reg field, the ModRegR/M byte is not used
  - Immediate values and offset values are encoded in *little-endian* order!
  - Stop.
- Step 3: (there is a ModRegR/M byte) Otherwise, set the Reg field of next byte according to the note

Encoding (cont.)

- Step 4: Match the addressing mode of R/M from the "ModR/M Byte Specification" table
  - If there is one operand, R/M = that operand
  - Else if second operand is an immediate, R/M = first operand
  - Else if both operands are registers, R/M = second operand
  - Else if first operand is in memory, R/M = first operand
  - Else R/M = second operand
- Step 5: Fill in Mod field based on this table, and fill in R/M field

Encoding (cont.)

- Step 6: If based-indexed mode not used, follow ModRegR/M with displacement value (if any), followed by immediate value (if any)
  - Stop.
- Step 7: Otherwise, match the exact form of the index with the table "SIB Byte"
  - Set the SS field according to the table
  - Set the Index field accroding to the table
  - Set the Base field according to the base register used

Encoding (cont.)

- Step 8: Follow SIB with displacement value (if any), followed by immediate value (if any)
  - And stop!

Conditional Jumps

- The operand of a conditional jump is the displacement from the address of the *next instruction after the jump* to the target instruction address
  - Positive displacement if a "forward" jump
  - Negative displacement if a "backwards" jump
- The displacement is a 32-bit value (for us)